

# Fault Diagnosis Based On Causal Computations

Albert Rosich, Erik Frisk, Jan Åslund, Ramon Sarrate and Fatiha Nejari

**Abstract**—This work focuses on residual generation for model-based fault diagnosis. Specifically, a methodology to derive residual generators when non-linear equations are present in the model is developed. A main result is the characterization of computational sequences that are particularly easy to implement as residual generators and that take causal information into account. An efficient algorithm, based on the model structure only, that finds all such computational sequences, is derived. Further, fault detectability and fault isolability performance depend on the sensor configuration. Therefore, another contribution is an algorithm, also based on model structure, that places sensors with respect to the class of residual generators that take causal information into account. The algorithms are evaluated on a complex, highly non-linear, model of a fuel cell stack system. A number of residual generators are computed that are, by construction, easy to implement and provide full diagnosability performance predicted by the model.

**Index Terms**—Fault diagnosis, causal computations, sensor placement, fuel cell stack system.

## I. INTRODUCTION

INDUSTRIAL processes can be affected by faults having a serious impact on operation when not promptly detected and identified. In order to diagnose these faulty behaviors, efficient diagnosis systems are of great importance for modern industries. Over the last three decades, the growing demand for safety and reliability has drawn significant research in fault detection and diagnosis based on a model of the system [1], [2], [3], [4].

Most approaches for model-based fault diagnosis rely on consistency checking. A comparison between the observed behavior and a model of the process is performed by means of a set of residual generators, which are designed by exploiting the redundancy in the model of the system. Fault detection is achieved when a residual generator is triggered upon the occurrence of a fault. Fault isolation is then performed by inferring the triggering pattern of a set of residuals. Most diagnosis systems deployed in industry are still based on quite basic techniques such as variable limit checking and there is a potential to increase diagnosis performance by using more advanced methods.

Many methods are difficult to use for industrial systems since the models typically include non-linearities such as

lookup-tables, saturations, hysteresis functions. There exist methods for dealing with such models [5], [6], [7], but they can often be practically infeasible. For example, methods based on variable elimination suffer from severe complexity problems and Gröbner basis techniques fail for even moderately sized systems [8, p. 108]. Another example is observer based techniques, as in [7], where analytical solutions to a non-linear partial-differential equation are needed in the design. Although theoretically sound, the design procedure is often not possible for industrial models, due to the size and complexity of the model equations.

One possible solution is to rearrange the model equations so that all variables can be computed using back substitution. However, this would require that parts of the model with redundancy can be rearranged into a triangular form which is a severe limitation on the class of models that can be used.

The main contribution of this paper is a method, placed somewhere in between the simple substitution approach and the more general techniques that rely on complex analytical computations. The computation of the residual is here decomposed into either linear sub-problems, which are easy to solve, or non-linear problems with a structure that allows a simple back substitution. To identify these sub-problems, a structural representation of the system is used together with a causal interpretation. A novelty with this paper is the extension of previous approaches [6] with a systematic treatment of linear and non-linear variables, where the non-linear are separated into causal and non-causal variables. Based on the generated residuals, basic techniques from consistency based diagnosis are used to perform the fault isolation. See for example [4] for basic fault isolation algorithms and [9] for how to integrate residual generation with such techniques.

Fault diagnosis relies on process observations, which are usually measured with sensors. Hence, the efficiency of a diagnosis system critically depends on the location of the sensors. For many systems there exists a great number of possible candidate residual generators [10], which means that the restriction on the class of residual generators may not severely limit the detection and isolation performance of a designed diagnosis system. For this reason, an interesting question is which sensors to use in order to achieve a given diagnosis specification using this class of residual generators. There exist some results devoted to sensor placement for diagnosis using graph tools. In [11], a digraph representation of the relationship between sensors and faults of the process is used as a basis for the sensor-location problem. In [12], an algorithm is developed for placing sensors but limited to linear differential-algebraic systems. In [13], structural analysis is applied but limited to linear structured systems. In [14], structural analysis is applied to non-linear systems but the

A. Rosich is in the Spanish project WATMAN (ref. DPI2009-13744) with the Institut de Robòtica i Informàtica Industrial (CSIC-UPC), Llorens i Artigas 4-6, 08028 Barcelona, Spain,

R. Sarrate and F. Nejari are with the Department of Automatic Control, Universitat Politècnica de Catalunya, Rambla de Sant Nebridi, 10, 08222 Terrassa, Spain (e-mail: albert.rosich@upc.edu; ramon.sarrate@upc.edu; fatiha.nejari@upc.edu)

E. Frisk and J. Åslund are with the Department of Electrical Engineering, Linköping University, 581 83 Linköping, Sweden (e-mail: frisk@isy.liu.se; jaasl@isy.liu.se).

method requires the previous computation of the complete set of redundant sub-models, which is a highly inefficient task for large scale complex systems. Lastly, in [15], a more efficient algorithm is developed which does not require this previous computation. However, existing techniques based on structural analysis give only best case results when applied to non-linear systems. This drawback is alleviated in this present work, by formulating a sensor placement algorithm which takes into consideration the causal computability in the residual generation. Previous works, as [15], neither focus on residual generation nor handle causal variables.

This paper is organized as follows. In Section II, the problem to be solved is motivated and defined. In Section III, the guidelines to handle causalities within a structural model are presented and algorithms to determine the computable part of the model are proposed. Next, in Section IV, the previous algorithms are applied to determine the fault diagnosis properties of the system. The sensor placement problem is addressed in Section V, whereas Section VI deals with the computation of the sub-models which are used for residual implementation. Finally, in Section VII, the whole methodology is applied to a fuel cell stack system [16] where main advantages of the proposed approach are illustrated. The fuel cell system model is complex, involving a wide range of non-linear equations including look-up tables, piecewise polynomial functions, non-linear dynamic equations, etc. The model also covers a wide range of operating points.

## II. PROBLEM BACKGROUND AND MOTIVATION

In model based diagnosis, consistency is checked by using a set of sub-models with redundancy. One approach is to analyze the model structure and find *minimal* sub-models with redundancy. These are the smallest sets of equations that can be used to compute a residual. The name given to minimal sub-models with redundancy depends on the approach, for example *analytical redundancy relations*, ARR [17], *minimal structurally overdetermined set*, MSO [10], *testable sub-system*, TSS [18] and *minimal evaluation chain*, MEC [19].

A residual generator will here be realized from a minimal redundant sub-model by computing the internal unknown variables through a convenient manipulation of sub-model equations and checking consistency in a redundant equation. This concept is known as a causal interpretation of the computability [6]. For instance, in [18] causality is taken into account in the computation of the set of redundant sub-models whereas in [20] causality is considered in derivative and integral computations. The causal interpretation can be represented by a directed bi-partite graph that shows how the internal values can be computed from the equations (value propagation) in every redundant sub-model. However, to guarantee that the residual can be generated by using non-linear equations, the structural model framework must be adapted in order to handle causal computability.

To illustrate residual generation based on a causal interpretation, i.e., a computable sequence for the unknown variables, consider the model (1). It consists of three equations ( $e_1$ ,  $e_2$  and  $e_3$ ), where  $y_1$  and  $y_2$  are known variables, and  $x_1$  and  $x_2$  are unknown variables.

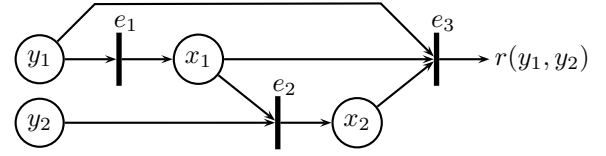


Fig. 1. Computation sequence.

$$\begin{aligned} e_1 : x_1 &= h_1(y_1) \\ e_2 : x_2 &= h_2(x_1, y_2) \\ e_3 : h_3(x_1, x_2, y_1) &= 0 \end{aligned} \quad (1)$$

A corresponding computation sequence for the unknown variables can be constructed (see Fig. 1). Equation  $e_1$  is used to compute variable  $x_1$  and equation  $e_2$  to compute variable  $x_2$  and it is then straightforward to propagate the values to compute the residual as in (2).

$$r(y_1, y_2) = h_3(h_1(y_1), h_2(h_1(y_1), y_2), y_1) \quad (2)$$

where,

$$\begin{cases} r(y_1, y_2) \simeq 0 & \text{means that there is consistency} \\ r(y_1, y_2) \not\simeq 0 & \text{means that there is no consistency} \end{cases} \quad (3)$$

Using this procedure to design residual generators in complex systems gives an intuitive idea on how a residual can be computed. However, solving a certain variable in a non-linear equation could be a hard task, or even impossible, which ultimately poses restrictions on the residual generator design. This means that not all matchings can be used to design a residual generator as in the example. This will lead to a restricted set of residuals and consequently a restricted set of corresponding minimal sub-models with redundancy.

Specifically, residuals generators that depend on sub-models that imply the inverse computation of non-invertible functions will be excluded. Furthermore, equation subsets that involves loops, both algebraic and differential, in the computation sequence will be excluded as well. Therefore, no nonlinear solving tools will be needed and the residual computation will be ensured. On the other hand, to keep the simplicity of the approach and at the same time reduce the restrictiveness, sub-models including linear loops will not be excluded, since solving linear equations is not a complex task. A consequence of this extension is that existing structural methods for finding sub-models and computational sequences have to be modified. A main contribution of this work is that the design of the diagnosis systems and the sensor placement analysis take into account which methodology is used to compute residuals.

## III. CAUSAL FRAMEWORK

### A. Causal Structural Model

To determine when a redundant sub-model can be used to generate a residual, using a computation sequence, some information on how variables can be computed in each equation is required. In non-linear equations, unknown variables can not always be computed as a function of the others, for instance when non-invertible functions are regarded. This leads to the following definition:

**Definition 1 (Causally computable variable):** Let  $h(\mathbf{x}) = 0$  be an equation of the model. Variable  $x_i \in \mathbf{x}$  is causal in  $h$ , if  $x_i$  can be computed using  $h$ , assuming that the remaining variables,  $\mathbf{x} \setminus x_i$ , are known. We say that there is a causal relation between  $x_i$  and  $h$ .

From Definition 1 it follows that equation  $h$  can never be used in the computation sequence to compute non-causal variables. Furthermore, as mentioned before, causal variables that are involved in non-linear loops are not computable in the computation sequence. For instance, the two expressions in (4) are used to calculate the compressor efficiency,  $\eta$ , and the compressor torque,  $\tau$ , in the fuel cell stack model. Assume that the compressor pressure,  $p$ , the angular speed,  $\omega$ , the atmospheric temperature,  $T_{atm}$ , and the compressor torque,  $\tau$  are known or measured variables, whereas the efficiency,  $\eta$ , and the air flow,  $W$ , are unknown variables. Constants  $C_p$  and  $\gamma$  are known system parameters.

$$\eta = \text{LookupTable}(p, W) \quad (4a)$$

$$W = \tau \left( C_p \frac{T_{atm}}{\eta \cdot \omega} (p^{\frac{\gamma-1}{\gamma}} - 1) \right)^{-1} \quad (4b)$$

Note that in the first equation, a look-up table is used to calculate the compressor efficiency from the air flow and the compressor pressure. Thus neither the pressure nor the air flow can be computed using this expression. According to Definition 1, the unknown variable  $\eta$  is causal in the first equation, whereas  $W$  is not. In the second equation, both unknown variables,  $\eta$  and  $W$ , are causal variables.

This is a well-constrained set of equations and there is a causal relation between unknown variables and equations, i.e. equation (4a) can be used to compute  $\eta$  and equation (4b) can be used to compute  $W$ . However, it is not possible to compute the unknown variables by forward value propagation since both unknown values must be computed at the same time.

This kind of structure is known as an *algebraic loop*. There are several tools to compute unknown variables in an algebraic loop (e.g., numeric solvers, non-linear optimization, tearing techniques), but the solution is not always ensured and the computation cost can be large. In this work, a conservative approach consisting in rejecting all non-linear algebraic loops is adopted. On the other hand, linear algebraic loops are easier to handle as long as algebraically independent coefficients are assumed. Thus, in this work, algebraic loops involving linear variables will be accepted in a computation sequence. This motivates Definition 2.

**Definition 2 (Linear Variable):** Let  $h(\mathbf{x}) = 0$  be an equation of the model. A set of variables  $\mathbf{x}_l \subseteq \mathbf{x}$  is linear in  $h$  if  $h$  can be arranged as  $\mathcal{L}(\mathbf{x}_l) + g(\mathbf{x} \setminus \mathbf{x}_l) = 0$  and  $|\mathbf{x}_l| > 1$ , where  $\mathcal{L}$  is a linear function. We say that there is a linear relation between  $\mathbf{x}_l$  and  $h$ .

Note that considering one single variable as a linear variable in an equation is not necessary. Linear relations are meant to be considered for identifying linear algebraic loops, and one single variable never forms a loop. Thus, linear relations are considered when two or more linear variables appear in the same equation.

To exemplify a linear algebraic loop, consider the electric motor equations from the air compressor model (5), where all variables are linear. The compressor voltage,  $v$ , and the compressor torque  $\tau$  are known variables, whereas the compressor current,  $i$ , and the angular speed,  $\omega$ , are unknown variables.

$$R \cdot i + k_v \cdot \omega = v \quad (5a)$$

$$k_p \cdot i - J \frac{d\omega}{dt} - B \cdot \omega = \tau \quad (5b)$$

Since both equations are linear, the unknown variables,  $i$  and  $\omega$  can be easily computed, in spite of the existence of an algebraic loop:

$$\begin{pmatrix} i \\ \omega \end{pmatrix} = \begin{pmatrix} R & k_v \\ k_p & -(Jp + B) \end{pmatrix}^{-1} \begin{pmatrix} v \\ \tau \end{pmatrix} \quad (6)$$

where  $R$ ,  $k_v$ ,  $k_p$ ,  $J$  and  $B$  are model parameters and  $p$  is the differentiation operator.

The structure of a model can be formalized as a bipartite graph  $G(M, X, A)$ , where  $M = \{\dots, e_i, \dots\}$  is the set of model equations,  $X = \{\dots, x_j, \dots\}$  the set of unknown variables and  $A$  the set of edges, such that  $(e_i, x_j) \in A$  if equation  $e_i \in M$  depends on the variable  $x_j \in X$ . Information on causal and linear relations can be well fitted in the structural model by a partition of the set of edges  $A = A_L \cup A_\times \cup A_\Delta$  where, according to the previous definitions:

- $A_L$  is a subset of edges such that  $x_j$  is a linear variable in  $e_i$ .
- $A_\times$  is a subset of edges such that  $x_j$  is a causal but not linear variable in  $e_i$ .
- $A_\Delta$  is the remaining subset of edges, where  $x_j$  is a non-causal variable in  $e_i$ .

In the biadjacency matrix, edges in  $A_L$  are represented by an “L” symbol, edges in  $A_\times$  are represented by a “ $\times$ ” symbol and edges in  $A_\Delta$  are represented by a “ $\Delta$ ” symbol.

A central concept used frequently in the following sections is *matching* [21]. A matching is a set  $\Gamma$  of edges such that no two edges in  $\Gamma$  have common nodes. A matching can, in the context of structural models, loosely be interpreted as which variable is solved in which equation.

## B. Causal Computability

Given a structural model, there is a need to know whether a set of unknown variables can be computed when causal and linear relations are considered. Let  $G(M, X, A)$  be a structural model with  $A = A_L \cup A_\times \cup A_\Delta$ . First, for the sake of simplicity, assume that there are no linear variables, i.e.,  $A_L = \emptyset$ .

In the previous section it was exemplified how a residual can be computed using a computational sequence. To be able to do this in a general case, a necessary condition is that there exists a complete matching  $\mathcal{M}_G^X$  in  $X$ , such that

$$\mathcal{M}_G^X \subseteq A_\times \quad (7)$$

i.e., only casual variables are matched.

As pointed out in the previous subsection, only algebraic loops involving linear variables will be accepted. In the absence of linear variables, this means that the well-constrained

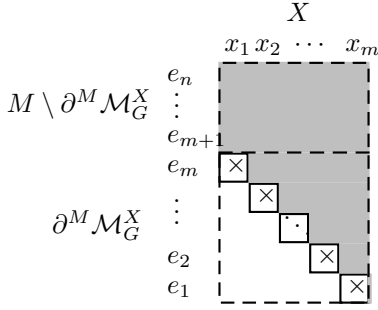


Fig. 2. Causally computable structure

subgraph  $G'(\partial^M \mathcal{M}_G^X, X)$  has no *Hall components* [22] with more than one equation, where  $\partial^M \mathcal{M}_G^X$  is the subset of equations in  $M$  incident to edges in  $\mathcal{M}_G^X$ .

Therefore, if a matching with such properties exists then the set of unknown variables,  $X$ , can be computed using the computation sequence without loops. Note that this means that the set of equations and the set of variables can be rearranged such that the biadjacency matrix has a triangular form with a diagonal of “ $\times$ ” symbols. Fig. 2 shows this pattern where all unknown variables can be evaluated.

Algorithm 1 searches for the set of variables that can be computed as causal variables. This is iteratively done by finding equations that only contain one causal variable,

$$e \in M : |\text{var}_X(e)| = 1 \wedge (e, \text{var}_X(e)) \in A_\times \quad (8)$$

where  $\text{var}_X(E)$  denotes the set of variables in  $X$  adjacent to the set of equations  $E$ . Note that

$$(e, \text{var}_X(e)) \in \mathcal{M}_G^X$$

according to (7) and (8). After finding equation  $e$  the graph is pruned and the algorithm continues searching for more equations until no more equation-variable pairs can be found.

---

**Algorithm 1**  $\mathcal{X}_C = \text{CausalVariable}(G(M, X, A))$

---

```

 $\mathcal{X}_C := \emptyset$ 
while  $\exists e \in M : |\text{var}_X(e)| = 1 \wedge (e, \text{var}_X(e)) \in A_\times$  do
   $X := X \setminus \text{var}_X(e)$ 
   $\mathcal{X}_C := \mathcal{X}_C \cup \text{var}_X(e)$ 
end while

```

---

Here, it is assumed that a set of variables can be solved if every variable can be matched with an equation using a causal edge and there are no algebraic loops, so any *Hall component* with more than one equation-variable pair is rejected.

Now, assume that also linear variables are considered, i.e.  $A = A_L \cup A_\times \cup A_\Delta$ . In Section III-A, it was discussed that a subset of linear variables can be solved in an algebraic loop. Therefore, the Dulmage-Mendelsohn decomposition can be applied to determine the subset of linear computable variables. The Dulmage-Mendelsohn decomposition [23], [24] defines a partition on the set of equations and the set of variables. This partition consists in the under-determined part, the just-determined part and the over-determined part, which contains the redundant equations.

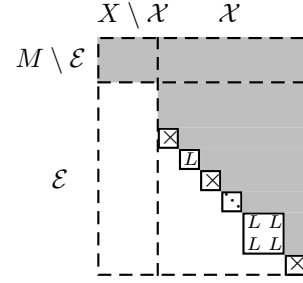


Fig. 3. Causal and linear computable decomposition

Since causal variables are determined by means of Algorithm 1, now we are only interested in finding the linear variables that can be computed. First, the set of equations  $E_L$  that depend on linear variables and no others is identified

$$E_L = \{e \in M : \forall x \in \text{var}_X(e), (e, x) \in A_L\} \quad (9)$$

Then, the set of linear computable variables  $\mathcal{X}_L$  is determined by applying the Dulmage-Mendelsohn decomposition to the equations subset  $E_L$ ,

$$\mathcal{X}_L = \text{var}_X(E_L^0) \cup \text{var}_X(E_L^+) \quad (10)$$

where  $E_L^0$  and  $E_L^+$  denote respectively the just-determined and the over-determined equation sets of the Dulmage-Mendelsohn decomposition in  $E_L$ . Note that this holds with the assumption that the linear coefficients are algebraically independent.

Algorithm 2 is developed to compute the set of linear computable variables  $\mathcal{X}_L$  according to (9) and (10).

---

**Algorithm 2**  $\mathcal{X}_L = \text{LinearVariable}(G(M, X, A))$

---

```

 $E_L := \{e \in M : \forall x \in \text{var}_X(e), (e, x) \in A_L\}$ 
 $\mathcal{X}_L := \text{var}_X(E_L^0) \cup \text{var}_X(E_L^+)$ 

```

---

The diagonal matching presented in Fig. 2 is now extended to include linear computable variables. The resulting structural decomposition is shown in Fig. 3 where the triangular form remains, but now the *Hall components* can include more than one variable, since linear loops are allowed. This decomposition is done by Algorithm 3, which iteratively alternates Algorithms 1 and 2, and finally returns the sub-graph  $G(\mathcal{E}, \mathcal{X}, A)$  which corresponds to the computable part.

From the discussion above, it is clear that the sub-graph  $G(\mathcal{E}, \mathcal{X}, A)$  contains the computable part of the model. Thus, all remaining equations,  $M \setminus \mathcal{E}$ , are not useful anymore, since they contain variables that can not be computed, i.e.,  $X \setminus \mathcal{X}$ .

---

**Algorithm 3**  $\{\mathcal{X}, \mathcal{E}\} = \text{ComputableSystem}(G(M, X, A))$

---

```

 $\mathcal{X} := \emptyset$ 
repeat
   $\mathcal{X}' := \text{CausalVariable}(G(M, X \setminus \mathcal{X}, A))$ 
   $\mathcal{X} := \mathcal{X} \cup \mathcal{X}'$ 
   $\mathcal{X}_L := \text{LinearVariable}(G(M, X \setminus \mathcal{X}, A))$ 
   $\mathcal{X} := \mathcal{X} \cup \mathcal{X}_L$ 
until  $\mathcal{X}' \cup \mathcal{X}_L = \emptyset$ 
 $\mathcal{E} := \{e \in M : \text{var}_X(e) \subseteq (\mathcal{X})\}$ 

```

---

Note that extracting the computation sequence given by the subgraph  $G(\mathcal{E}, \mathcal{X}, A)$ , decomposed as in Fig. 3, is straightforward since now the matching-diagonal establishes an interpretation of which equation to use to compute each variable.

#### IV. CAUSAL STRUCTURAL MODEL BASED DIAGNOSIS

According to the decomposition in Fig. 3 there exists, at least, one complete matching in  $G(\mathcal{E}, \mathcal{X}, A)$ . This means that there is no under-determined subset of equations in  $\mathcal{E}$ , i.e.,  $\mathcal{E}^- = \emptyset$ . Since the matching in Fig. 3 is complete in  $\mathcal{X}$ , it follows that the over-determined set of equations  $\mathcal{E}^+$  contains part of this diagonal matching, so variables in  $\mathcal{E}^+$  are computable. Now, fault diagnosis analysis can be performed on the over-determined part and the computation sequence can always be guaranteed to generate residuals. For extended information on the Dulmage-Mendelsohn decomposition applied to fault diagnosis see for example [6].

In this work, faults are defined as a subset of equations,  $F \subseteq M$ , since a relation between an equation and a fault can be easily established, i.e., a signal fault that affects an equation, or the assumption or support of an equation. In order to simplify the following theoretic development, only system faults and faults in the original sensor setup will be considered, i.e., no faults in additional sensors will be included in the sensor placement analysis.

##### A. Causal Structural Detectability

It is well known that the set of detectable faults can be defined from the over-constrained part [6]. A given set of faults  $F \subseteq M$  is structurally detectable if  $F \subseteq M^+$ . Analogously, causal (structural) detectability can be defined from the computable part of the model.

**Definition 3 (Causal (Structural) Detectability):** A fault  $f \in F$  is causally detectable in  $M$  if

$$f \in \mathcal{E}^+ \quad (11)$$

where  $\mathcal{E}$  is the computable part of  $M$ .

Algorithm 4 uses Algorithm 3 to find all causally detectable faults  $F_D$ , when computable sequences are taken into account. The inputs are a structural model  $G(M, X, A)$  and a set of faults,  $F \subseteq M$ .

---

**Algorithm 4**  $F_D = \text{CausalDetectability}(G(M, X, A), F)$

---

$\{\mathcal{X}, \mathcal{E}\} := \text{ComputableSystem}(G(M, X, A))$

$F_D := \mathcal{E}^+ \cap F$

---

##### B. Causal Structural Isolability

Isolability analysis is based on detectability conditions. According to [15], a fault  $f_i \in F$  is structurally isolable from  $f_j \in F$  if  $f_i$  is detectable in the sub-model  $M \setminus \{f_j\}$ . The same holds when causal computations are considered.

**Definition 4 (Causal (Structural) Isolability):** Given two causally detectable faults  $f_i, f_j \in F$ , fault  $f_i$  is causally isolable from fault  $f_j$  in  $M$  if

$$f_i \in \mathcal{E}_{f_j}^+ \quad (12)$$

TABLE I  
NON-SYMMETRIC ISOLABILITY EXAMPLE

	$x_1$	$x_2$
$f_1 \rightarrow e_1$	$\Delta$	$\Delta$
$f_2 \rightarrow e_2$	$\times$	$\times$
$e_3$	$\Delta$	$\times$
$e_4$		$\times$

where  $\mathcal{E}_{f_j}$  is the computable part of  $M \setminus \{f_j\}$ .

This means that for each causally detectable fault  $f \in F_D$ , there exists a set of faults  $F_I(f)$  that are causally isolable from  $f$ . Algorithm 5 uses Algorithm 4 to compute the causally isolable fault set for each causally detectable fault.

---

**Algorithm 5**  $F_I = \text{CausalIsolability}(G(M, X, A), F_D)$

---

**for each**  $f \in F_D$  **do**

$M_f := M \setminus \{f\}$

$F_I(f) := \text{CausalDetectability}(G(M_f, X, A), F_D)$

**end for**

---

Note that, here, the isolability relation between two faults is not symmetric, i.e., fault  $f_i$  is isolable from fault  $f_j$  does not imply that  $f_j$  is isolable from  $f_i$ . Since causal detectability depends on the causally computable sub-model, the symmetry property in the isolability relation (see [15]) is lost.

For instance, assume the following causal structural model represented in Table I, where faults  $f_1$  and  $f_2$  affect equation  $e_1$  and  $e_2$  respectively, and both are detectable  $F_D = \{e_1, e_2\}$ . Then, by applying Algorithm 5 we obtain that

$$\begin{aligned} F_I(e_1) &= \{e_2\} \\ F_I(e_2) &= \emptyset \end{aligned}$$

and it can be therefore concluded that  $f_2$  is isolable from  $f_1$  but the reverse does not hold (i.e., the symmetry property is not satisfied).

#### V. SENSOR PLACEMENT FOR CAUSAL STRUCTURAL MODEL BASED DIAGNOSIS

Given a set of equations, the subset of unknown variables that can be computed will depend on the set of installed sensors (i.e., known variables).

The main idea is to perform fault detectability and fault isolability analysis with all sensors installed. Under this setting, the set of detectable faults and the set of isolable faults will give an upper limit on the fault diagnosis specifications. Installing the same sensor more than once makes neither detectable a non-detectable fault nor isolable a non-isolable fault, except for faults in the installed sensors. Once maximum fault diagnosis specifications are known, the minimal set of sensors that satisfies these specifications is sought.

##### A. Maximum causal detectability and isolability specifications

The maximum detectability specification is ensured when all candidate sensors are installed. Therefore, it is straightforward to select those faults that can be detected from those that will never be, before the sensor placement analysis.

The set of candidate sensors can be defined as a subset of unknown variables,  $S \subseteq X$ . Each sensor has a corresponding sensor equation  $y = x$ , with  $y$  being the measurement signal and  $x \in S$  the measured variable. This equation has to be added to the model whenever the corresponding sensor is selected for installation. Note that adding this equation implies that  $x$  becomes a causal variable in the corresponding sensor equation. Given a sensor configuration  $S_k \subseteq S$ , the set of sensor equations is denoted by  $M_{S_k}$ .

Even if all sensors are added, there may be some sensors that can not be used to compute a residual. It is important to identify these sensors in order to exclude them from the sensor placement analysis. These sensors are characterized by the property that the corresponding sensor equation does not belong to the over-determined part of the computable subsystem. Therefore, it is possible to determine the set of useful sensors from

$$M_{S_d} = \mathcal{E}_S^+ \cap M_S \quad (13)$$

where  $\mathcal{E}_S$  is the computable part of the system with all sensors installed, i.e.,  $\{\mathcal{X}, \mathcal{E}_S\} := \text{ComputableSystem}(G(M \cup M_S, X, A))$ . Thus, the new set of candidate sensors,  $S_d \subseteq S$ , is now defined as all sensors such that their corresponding sensor equation belongs to  $M_{S_d}$ . Therefore, there is no need to further consider sensors from  $S \setminus S_d$  in the sensor placement analysis.

Now, the maximum causal detectability specification is computed by Algorithm 4 with the new set of sensors  $S_d$  added in the model:

$$F_{D_{max}} = \text{CausalDetectability}(G(M \cup M_{S_d}, X, A), F)$$

The maximum causal isolability specification is computed by Algorithm 5, with just the set of sensors  $S_d$  installed, and for those system faults that are detectable,  $F_{D_{max}}$ :

$$F_{I_{max}} = \text{CausalIsolability}(G(M \cup M_{S_d}, X, A), F_{D_{max}})$$

### B. Sensor placement algorithm

Once maximum causal detectability and isolability specifications are known, the sensor placement algorithm can be introduced. Algorithm 6 uses Algorithms 4 and 5 to search for the minimal set of sensors that satisfies them (i.e.,  $F_{D_{max}}$  and  $F_{I_{max}}$ ).

The algorithm starts each iteration by choosing the minimal set of sensors ( $S_k$ ) not already chosen. The sensor set cost or cardinality are different criteria that could be used to determine the minimal set of sensors. Then, the algorithm computes causal detectability using the chosen sensor set and tests whether maximum causal detectability is achieved. If so, the same is done with causal isolability and maximum causal isolability. When both, maximum causal detectability and isolability, are achieved the solution ( $S_{min}$ ) is returned.

In Section V-A, it has been shown that the algorithm will terminate since the set of candidate sensors,  $S_d$ , is one admissible set, that fulfills the specifications.

---

#### Algorithm 6 $S_{min} = \text{CausalSensorPl}(M, S_d, F_{D_{max}}, F_{I_{max}})$

---

```

1:  $S_{min} := \emptyset$ 
2: repeat
3:    $S_k :=$  the minimal subset not previously tested from  $S_d$ 
4:    $F_D :=$ 
       CausalDetectability( $G(M \cup M_{S_k}, X, A), F_{D_{max}}$ )
5:   if  $F_D = F_{D_{max}}$  then
6:      $F_I :=$  CausalIsolability( $G(M \cup M_{S_k}, X, A), F_{D_{max}}$ )
7:     if  $F_I = F_{I_{max}}$  then
8:        $S_{min} := S_k$ 
9:     end if
10:  end if
11: until  $S_{min} \neq \emptyset$ 
```

---

## VI. CAUSALLY COMPUTABLE MSO SET GENERATION

Finding redundant sub-systems for diagnosis is an important topic in the field of diagnosis based on structural models. There are several works devoted to this issue [17], [10], [18], [19]. An efficient algorithm that computes the complete set of MSO (Minimal Structural Overdetermined) sets was published in [10]. An MSO set is a subset of model equations that is structurally overdetermined and no proper subset is overdetermined. Furthermore, an MSO set can be used to implement a residual generator. A modification of this algorithm is presented in this section. It consists in only computing those MSO sets that can be used to generate a residual by means of the computation sequence. This kind of MSO set is called *causally computable MSO set*. Therefore, a causally computable MSO set is an MSO set that contains a causally computable structure, which means that it can be decomposed as in the  $(\mathcal{E}, \mathcal{X})$  structure depicted in Fig. 3. The extension of the algorithm in [10], that computes the complete set of MSO sets, is presented in Algorithm 7.

---

#### Algorithm 7 $\mathcal{M} = \text{findCCMSO}(G(M, X, A), R)$

---

```

1:  $\mathcal{M} := \emptyset$ 
2:  $\{\mathcal{X}, \mathcal{E}\} := \text{ComputableSystem}(G(M, X, A))$ 
3: if  $(M \setminus \mathcal{E}^+) \cap R = \emptyset$  then
4:    $M := \mathcal{E}^+$ 
5:   if  $\bar{\varphi}(M) = 1$  then
6:      $\mathcal{M} := \mathcal{M} \cup \{M\}$ 
7:   else
8:     while  $R \not\subseteq M$  do
9:       Select an  $e \in M \setminus R$ 
10:       $E := M \setminus (M \setminus \{e\})^+$ 
11:      if  $E \cap R = \emptyset$  then
12:         $R := R \cup E$ 
13:         $M' := M \setminus E$ 
14:         $\mathcal{M} := \mathcal{M} \cup \text{findCCMSO}(G(M', X, A), R)$ 
15:      else
16:         $R := R \cup E$ 
17:      end if
18:    end while
19:  end if
20: end if
```

---

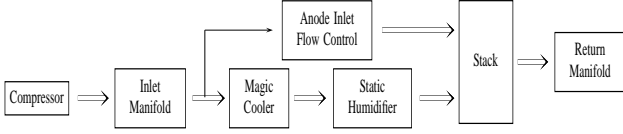


Fig. 4. Fuel Cell System scheme

The algorithm computes the set of causal MSO sets by iteratively removing equations. The set  $R$  is a set of equations that are not allowed to be removed to avoid finding the same MSO set more than once. Algorithm 7 is initially called with the entire model  $G(M, X, A)$  and  $R = \emptyset$ . The condition to ensure that an MSO set has been found is that  $\bar{\varphi}(M) = 1$  (step 5), where  $\bar{\varphi}(M)$  is the structural redundancy degree, which is defined by

$$\bar{\varphi}(M) = |M| - |\text{var}_X(M)| \quad (14)$$

The extension, compared to the algorithm in [10], mainly consists of steps 2 to 4. The set  $M \setminus \mathcal{E}^+$  is the set of equations that are removed in step 4 and, in step 3, it is ensured that no equations in  $R$  are removed. The remaining part of Algorithm 7 is equivalent to the MSO sets generation algorithm in [10]. The original algorithm in [10] finds all possible MSO sets. Since the set of causally computable MSO sets is a subset of all the MSO sets, Algorithm 7 finds all possible causally computable MSO sets,  $\mathcal{M}$ .

## VII. APPLICATION TO THE FUEL CELL STACK SYSTEM

Fuel cell devices are receiving much attention in the last decade as good candidates for clean electricity generation. Here, we will use a fuel cell based system to apply the presented diagnosis approach.

### A. System description

The model of the Polymer Electrolyte Membrane (PEM) fuel cell stack (FCS) system used in this work was proposed in [25], and further information can be found in [16]. The benchmark is widely accepted in the control community as a good representation of the behavior of a fuel cell system. The model (see Fig. 4) includes a detailed description of the air compressor, the inlet and return cathode manifolds, the static air cooler, the static humidifier, the hydrogen flow and the PEM fuel cell stack. The fuel cell stack model is further decomposed into four main subsystems: stack voltage, cathode flow, anode flow and membrane hydration. In the model, it is assumed that the temperature is known and constant since its dynamic behavior is much slower than that of the rest of the model.

In [26] there is a model of the FCS with 116 equations. The equations describe, in great detail, the physics and chemistry in the components. However, the model comprises non-linear relations, for example non-linear algebraic equations, piecewise polynomial functions, function maps, and look-up tables, which makes the proposed approach suitable.

A set of seven faults  $F$  has been selected for this evaluation study. Each fault affects one, and only one, equation, for

TABLE II  
SYSTEM FAULTS,  $F$

Fault	Description	Involved equation
$f_1$	Electrical fault in the compressor motor	$e_1$
$f_2$	Mechanical fault in the compressor motor	$e_2$
$f_3$	Compressor fault. The relation described by the <i>compressor map</i> does not hold	$e_{11}$
$f_4$	Air leak in the <i>inlet manifold</i>	$e_{13}$
$f_5$	Humidifier fault. Output humidity does not follow desired humidity	$e_{29}$
$f_6$	Cathode <i>return manifold</i> fault	$e_{33}$
$f_7$	Exit cathode fault	$e_{75}$

TABLE III  
CANDIDATE SENSORS

Variable	Description
$\omega_{cp}$	Compressor motor speed
$i_{cp}$	Compressor motor current
$T_{cp,out}$	Compressor output air flow temperature
$p_{cp,out}$	Compressor output pressure
$T_{im,out}$	Inlet manifold output air flow temperature
$p_{im,out}$	Inlet manifold output pressure
$T_{mc,out}$	Magic cooler output air flow temperature
$p_{mc,out}$	Magic cooler output pressure
$T_{sh,out}$	Static humidifier output air flow temperature
$p_{sh,out}$	Static humidifier output pressure
$T_{om,out}$	Outlet manifold output air flow temperature
$p_{an,in}$	Stack anode input pressure
$p_{ca,in}$	Stack cathode input pressure
$p_{st,ds}$	Stack downstream pressure
$v_{st}$	Stack voltage
$T_{an,out}$	Stack anode output flow temperature
$p_{an,out}$	Stack anode output pressure
$W_{ca,out}$	Stack cathode output flow
$T_{ca,out}$	Stack cathode output flow temperature
$p_{ca,out}$	Stack cathode output pressure

example by changing a parameter or a variable. Table II summarizes the faults considered in this work. Other faults could be easily included in this set, that should be related to their corresponding model equations. Another assumption is that only single faults are allowed. This means that two or more faults can not occur in the system at the same time.

Furthermore, there is a set of already known variables: the compressor voltage ( $v_{cm}$ ) and stack current ( $i_{st}$ ) since they are needed for control purposes, the desired temperature ( $T_{des}$ ) and humidity ( $\phi_{des}$ ), both set-points, the stack temperature ( $T_{st}$ ) and all the ambient variables (pressure  $p_{amb}$ , temperature  $T_{amb}$  and humidity  $\phi_{amb}$ ). All these variables are excluded from the sensor placement problem.

### B. Sensor Placement for the FCS System

A set of 20 candidate sensors for installation has been considered for this benchmark. Table III briefly describes them. All these physical quantities can be easily measured by standard sensors. Other physical quantities such as humidity or mass have not been considered since measurement of those quantities usually involve complex and expensive sensors.

Following the methodology introduced in Section V-A, maximum causal detectability and isolability specifications are sought. The conclusion is that all faults can be detected and isolated, under the assumption that all candidate sensors are

installed. Thus,  $F_{D_{max}} = F$  and  $F_{I_{max}}(f_i) = F \setminus \{f_i\}$  for  $i = \{1, \dots, 7\}$ .

Moreover, the sensor equation corresponding to sensor  $v_{st}$  is not in the computable over-determined part. So, the effective set of candidate sensors becomes  $S_d = S \setminus \{v_{st}\}$ . This is due to the fact that there is only one equation (15) that depends on variable  $v_{st}$  and not all the variables involved in this equation can be causally computed.

$$e_{79} : v_{st} = h(i_{st}, p_{ca,out}, p_{O_2}, T_{st}, \lambda_m, p_{H_2}) \quad (15)$$

Now, the minimal cardinality sensor placement for diagnosis is solved. Applying Algorithm 6, six possible solutions are obtained. All sensor configurations are equivalent, involving 8 sensors as follows:

$$\begin{aligned} S_{min_1} &= S_{base} \cup \{p_{cp,out}, p_{ca,in}\} \\ S_{min_2} &= S_{base} \cup \{p_{cp,out}, p_{sh,out}\} \\ S_{min_3} &= S_{base} \cup \{p_{mc,out}, p_{ca,in}\} \\ S_{min_4} &= S_{base} \cup \{p_{mc,out}, p_{sh,out}\} \\ S_{min_5} &= S_{base} \cup \{p_{im,out}, p_{ca,in}\} \\ S_{min_6} &= S_{base} \cup \{p_{im,out}, p_{sh,out}\} \end{aligned}$$

with

$$S_{base} = \{\omega_{cp}, i_{cp}, T_{im,out}, p_{st,ds}, W_{ca,out}, p_{ca,out}\}$$

Sensor configuration  $S_{min_1}$  is finally adopted for the FCS system.

### C. Causal MSO Sets Generation for the FCS System

Once the set of sensors to achieve maximum diagnosis specification is known and their corresponding equations are introduced in the FCS structural model, Algorithm 7 is applied. It returns 323 causally computable MSO sets.

This will be compared to the number of MSO sets obtained under the assumption that all unknown variables were causal. Applying the original MSO set generation algorithm in [10], 219089 MSO sets are computed. Note that, for many of the MSO sets it is not possible to implement a residual generator due to computational problems.

Not all 323 corresponding residuals have to be implemented, but a reduced set. A subset of causally computable MSO sets will be selected such that it ensures maximum causal detectability and isolability of the fault set given in Table II. Following this criterion, 7 causally computable MSO sets have been selected for residual implementation. Table IV shows the fault sensitivity of these causally computable MSO sets. An MSO set is sensitive to a fault if the corresponding fault equation belongs to that MSO set. Note that all faults are causally detectable and isolable from each other, according to Definitions 3 and 4. The system and sensor equations that belong to each selected causally computable MSO set are described in (16).

$$\begin{aligned} MSO_1 &= \{\mathbf{e}_{75}, e_{124}, e_{125}, e_{126}\} \\ MSO_2 &= \{e_{31}, e_{32}, \mathbf{e}_{33}, e_{36}, e_{81}, e_{103}, e_{104}, e_{105}, e_{109}, \end{aligned}$$

TABLE IV  
RELATION BETWEEN MSO SETS AND SYSTEM FAULTS

	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	$f_6$	$f_7$
$MSO_1$							$\times$
$MSO_2$						$\times$	
$MSO_3$				$\times$	$\times$		
$MSO_4$			$\times$		$\times$		
$MSO_5$		$\times$		$\times$			
$MSO_6$		$\times$	$\times$				
$MSO_7$	$\times$						

$$\begin{aligned} &e_{116}, e_{124}, e_{125}\} \\ MSO_3 &= \{e_{12}, \mathbf{e}_{13}, e_{14}, e_{15}, e_{16}, e_{18}, e_{19}, e_{20}, e_{25}, e_{28}, \\ &\mathbf{e}_{29}, e_{82}, e_{83}, e_{84}, e_{85}, e_{87}, e_{88}, e_{89}, e_{90}, e_{94}, \\ &e_{95}, e_{96}, e_{97}, e_{98}, e_{100}, e_{101}, e_{102}, e_{114}, e_{121}, \\ &e_{122}, e_{123}, e_{126}, e_{119}, e_{121}, e_{122}, e_{126}\} \\ MSO_4 &= \{e_7, e_9, \mathbf{e}_{11}, e_{12}, e_{14}, e_{15}, e_{16}, e_{18}, e_{19}, e_{20}, \\ &e_{25}, e_{28}, \mathbf{e}_{29}, e_{82}, e_{83}, e_{84}, e_{85}, e_{86}, e_{87}, e_{88}, \\ &e_{89}, e_{94}, e_{95}, e_{96}, e_{97}, e_{98}, e_{100}, e_{101}, e_{102}, \\ &e_{114}, e_{119}, e_{121}, e_{122}, e_{123}\} \\ MSO_5 &= \{\mathbf{e}_2, e_3, e_4, e_6, e_8, e_9, \mathbf{e}_{13}, e_{14}, e_{15}, e_{16}, e_{18}, \\ &e_{19}, e_{83}, e_{84}, e_{85}, e_{86}, e_{87}, e_{90}, e_{119}, e_{120}, e_{121}, \\ &e_{122}, e_{126}\} \\ MSO_6 &= \{\mathbf{e}_2, e_3, e_4, e_6, e_7, e_8, e_9, \mathbf{e}_{11}, e_{14}, e_{15}, e_{16}, \\ &e_{18}, e_{19}, e_{83}, e_{84}, e_{85}, e_{86}, e_{87}, e_{119}, e_{120}, \\ &e_{121}, e_{122}\} \\ MSO_7 &= \{\mathbf{e}_1, e_{117}, e_{119}, e_{120}\} \end{aligned} \quad (16)$$

For instance, consider the causally computable MSO set  $MSO_2$  in (16). The system and sensor equations that belong to this MSO set are presented in (17). Note that variables  $T_{st}$ ,  $p_{om,ds}$ ,  $W_{ca,out}$  and  $p_{st,ds}$  are all known:  $W_{ca,out}$  and  $p_{st,ds}$  are measured (see Section VII-B),  $p_{om,ds}$  is the downstream outlet manifold pressure, which equals to the ambient pressure, and  $T_{st}$  equals the ambient temperature. Equations  $\{e_{31}, e_{32}, e_{33}, e_{36}\}$  belong to the *outlet manifold* component. The causal structural model corresponding to (17) is depicted in Table V. In equation  $e_{36}$ , a non-linear function computes the output manifold flow. This function is not invertible, so it cannot be used to compute any of the input variables, which is consistent with the row assigned to  $e_{36}$  in Table V. The same reasoning holds for each causally computable MSO set in (16). Note that the structure in Table V can be decomposed as in Fig. 3 (i.e., it is a computable structure).



TABLE V  
COMPUTABLE STRUCTURE OF  $MSO_2$

	$x$	$\dot{x}$	$W_{om,out}$	$T_{om,in}$	$p_{om,out}$	$T_{ca,out}$	$W_{om,in}$	$p_{om,ds}$	$T_{st}$	$p_{st,ds}$	$W_{ca,out}$
$e_{31}$	L			L							
$e_{32}$	L	L									
$e_{33}$		$\times$	L	$\times$			L				
$e_{36}$			$\times$	$\Delta$	$\Delta$			$\Delta$			
$e_{105}$				L		L				L	
$e_{116}$					L						
$e_{81}$						L			L		
$e_{104}$							L				L
$e_{103}$								$\times$			
$e_{109}$									$\times$		
$e_{124}$										$\times$	
$e_{125}$											$\times$

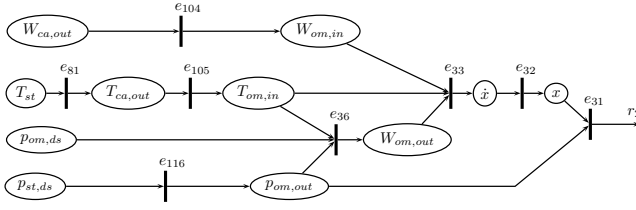


Fig. 5. Computation sequence from  $MSO_2$

$$\begin{aligned}
 e_{31} : p_{om,out} &= x \\
 e_{32} : \dot{x} &= \frac{dx}{dt} \\
 e_{33} : \dot{x} &= T_{om,in} \frac{R}{V_{om}} (W_{om,in} - W_{om,out}) \\
 e_{36} : W_{om,out} &= \text{NonlinearNozzle}(p_{om,out}, p_{om,ds}, T_{om,in}) \\
 e_{81} : T_{ca,out} &= T_{st} \\
 e_{103} : p_{om,ds} &= 1 \text{ atm} \\
 e_{104} : W_{om,in} &= W_{ca,out} \\
 e_{105} : T_{om,in} &= T_{ca,out} \\
 e_{109} : T_{st} &= 353 K \\
 e_{116} : p_{om,out} &= p_{st,ds} \\
 e_{124} : p_{st,ds} &= p_{st,ds_{measured}} \\
 e_{125} : W_{ca,out} &= W_{ca,out_{measured}}
 \end{aligned} \tag{17}$$

#### D. FCS System Diagnosis

Once the set of causally computable MSO sets is obtained, residuals can be easily implemented following the computation sequence. Note that the diagonal in the computable decomposition (see Fig. 3) shows how all variables can be computed. Therefore, implementing the computation sequence for a causally computable MSO set is straightforward. For instance, consider  $MSO_2$  shown in the previous subsection. The corresponding computation sequence is depicted in Fig. 5. Here sensor equations  $\{e_{124}, e_{125}\}$  and constant assignments  $\{e_{103}, e_{109}\}$  have been omitted in order to make the figure more readable. This computation sequence corresponds to the evaluation of residual  $r_2$ .

To test the residual  $r_2$ , fault  $f_6$  has been simulated by an abrupt change of parameter  $V_{om}$  in equation  $e_{33}$ , from nominal value  $5 \cdot 10^{-3} m^2$  to  $4.5 \cdot 10^{-3} m^2$ , at time 15 seconds. As long as fault  $f_6$  does not occur,  $r_2 \simeq 0$  (i.e., assuming model

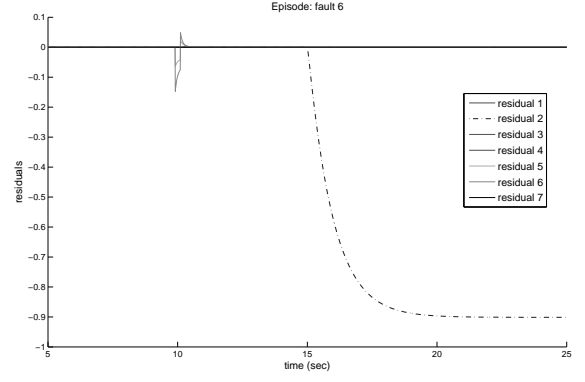


Fig. 6. Normalized residuals response for fault 6

inaccuracies). However, when fault  $f_6$  occurs, equation  $e_{33}$  does not hold and consequently  $r_2 \neq 0$ , signaling the fault (i.e., fault detection). Furthermore, according to (16),  $r_2$  is the only residual which is sensitive to fault  $f_6$ . This is consistent with Table IV, and implies that a violation of residual  $r_2$  indicates that fault  $f_6$  has occurred (i.e., fault isolation).

Fig. 6 shows the residuals response corresponding to fault  $f_6$  episode (normalized in the interval  $[-1, 1]$ ). At time 10 seconds, some residuals are slightly affected due to a change of the system operating point. However, at time 15 seconds the residual  $r_2$  is clearly affected by the fault, whereas the other residuals are not.

The same procedure could be followed to verify the performance of the residuals corresponding to the remaining MSO sets in (16). Remind that working with structural models, only best case results are obtained. This means that the residual sensitivities shown in Table IV will depend on the fault magnitude. Furthermore, a proper residual conditioning (i.e., filtering, thresholding, etc.) should be done. However, this topic is outside the scope of this work.

#### E. Comparison with the Non-Causal Approach

These results are compared to the solution of the minimal sensor placement problem when causal computability is not addressed. This can be accomplished by applying Algorithm 6 to a causal structural model of the fuel cell benchmark under the assumption that every unknown variable is a linear variable. Under this hypothesis, the minimal sensor configuration involves fewer sensors:  $\{i_{cp}, p_{st,ds}\}$ .

For this particular solution, there exist 7 useful MSO sets for causal fault detectability and isolability. The table of fault sensitivities of these 7 MSO sets is shown in Table VI. Note that all the faults are causally detectable and fully isolable. However, implementing residual generators with this fault sensitivity pattern is not a trivial task since all faults except one have to trigger the residual.

Furthermore, analyzing  $MSO_1^{nc}$  in Table VI, it could be seen that most of system equations are contained in this MSO:  $MSO_1^{nc} = M \setminus \{e_1, e_{34}, e_{35}, e_{79}, e_{106}\}$ , with  $M$  being the set of 116 system equations. Computing the unknown variables of this MSO set entails solving a non-linear system of equations.

TABLE VI  
RELATION BETWEEN NON-CAUSAL MSO SETS AND SYSTEM FAULTS

	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	$f_6$	$f_7$
$MSO_1^{nc}$		x	x	x	x	x	x
$MSO_2^{nc}$	x		x	x	x	x	x
$MSO_3^{nc}$	x	x		x	x	x	x
$MSO_4^{nc}$	x	x	x		x	x	x
$MSO_5^{nc}$	x	x	x	x		x	x
$MSO_6^{nc}$	x	x	x	x	x		x
$MSO_7^{nc}$	x	x	x	x	x	x	

A search through all choices of a redundant equation in this MSO set reveals that a non-linear algebraic loop of size at least 49 needs to be solved. Similar conclusions can be drawn for the other MSO sets.

For these reasons, implementing residual generators in non-linear large models by means of the computation sequence is often not possible when causal computations are not taken into account.

### VIII. CONCLUSIONS

Structural methods are often used to find suitable sets of equations that can be used to design residual generators. If invertability properties of the model are not taken into account in the analysis, then the resulting residual generators may include non-linear systems of equations that need to be solved either analytically or by using numerical techniques.

One way to avoid solving non-linear systems of equations is to take into account causal information and look for residual generators where the unknown variables are computed using direct back substitution. Such back substitution solutions are only possible under strong requirements on the model structure. The basic idea in this paper is to extend back substitution techniques to allow linear loops. This alleviates the model constraints but does not significantly increase the computational complexity of the residual generators. A main contribution of this paper is a structural framework and algorithms to identify sets of model equations where the unknown variables can be solved either by back substitution or by solving linear loops. In addition, a sensor placement algorithm has been developed that selects which sensors to include in order to meet a given diagnosability requirement under the assumption that all residual generators are designed using the extended back substitution approach.

A case study of a fuel cell stack system is used to illustrate the approach. With no causal restrictions on the residual generator, less sensors are needed to meet the diagnosability requirement, but this comes at the price of having to solve large non-linear systems of equations. Utilizing the causal information in the model results in a set of residual generators, fulfilling the isolability requirements, that can easily be computed by a simple back substitution approach and/or by solving linear loops.

### REFERENCES

[1] J. Gertler, *Fault Detection and Diagnosis in Engineering Systems*. New York: Marcel Dekker, Inc., 1998.  
[2] R. J. Patton, P. M. Frank, and R. N. Clark, *Issues of Fault Diagnosis for Dynamic Systems*. Springer, 2000.

[3] R. Reiter, "A theory of diagnosis from first principles," *Artificial Intelligence*, vol. 32, no. 1, pp. 57–95, 1987.  
[4] J. de Kleer and B. C. Williams, "Diagnosing multiple faults," *Artificial Intelligence*, vol. 32, no. 1, pp. 97–130, 1987.  
[5] M. Staroswiecki and G. Comtet-Varga, "Analytical redundancy relations for fault detection and isolation in algebraic dynamic systems," *Automatica*, vol. 37, no. 5, pp. 687–699, 2001.  
[6] M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki, *Diagnosis and Fault-Tolerant Control*, 2nd ed. Springer, 2006.  
[7] C. De Persis and A. Isidori, "A geometric approach to nonlinear fault detection and isolation," *IEEE Transactions on Automatic Control*, vol. 46, no. 6, pp. 853–865, 2001.  
[8] D. Cox, J. Little, and D. O'Shea, *Ideals, varieties, and algorithms*, 2nd ed. Springer, 1996.  
[9] M.-O. Cordier, P. Dague, F. Levy, J. Montmain, M. Staroswiecki, and L. Travé-Massuyès, "Conflicts versus analytical redundancy relations: a comparative analysis of the model based diagnosis approach from the artificial intelligence and automatic control perspectives," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 34, no. 5, pp. 2163–2177, 2004.  
[10] M. Krysander, J. Åslund, and M. Nyberg, "An efficient algorithm for finding minimal over-constrained sub-systems for model-based diagnosis," *IEEE Trans. Syst., Man, Cybern. A*, vol. 38, no. 1, 2008.  
[11] R. Raghuraj, M. Bhushan, and R. Rengaswamy, "Locating sensors in complex chemical plants based on fault diagnostic observability criteria," *AIChE J.*, vol. 45, no. 2, pp. 310–322, Feb. 1999.  
[12] E. Frisk, M. Krysander, and J. Åslund, "Sensor placement for fault isolation in linear differential-algebraic systems," *Automatica*, vol. 45, no. 2, pp. 364–371, 2009.  
[13] C. Commault, J. M. Dion, and S. Y. Agha, "Structural analysis for the sensor location problem in fault detection and isolation," *Automatica*, vol. 44, no. 8, pp. 2074–2080, aug 2008.  
[14] A. Rosich, R. Sarrate, V. Puig, and T. Escobet, "Efficient optimal sensor placement for model-based FDI using and incremental algorithm," in *Proc. 46th IEEE Conference on Decision and Control*, New Orleans, USA, Dec. 12–14, 2007, pp. 2590–2595.  
[15] M. Krysander and E. Frisk, "Sensor placement for fault diagnosis," *IEEE Trans. Syst., Man, Cybern. A*, vol. 38, no. 6, pp. 1398–1410, 2008.  
[16] J. T. Pukrushpan, "Modeling and control of fuel cell systems and fuel processors," Ph.D. dissertation, Univ. of Michigan, Ann Arbor, Michigan, 2003.  
[17] L. Travé-Massuyès, T. Escobet, and X. Olive, "Diagnosability analysis based on component supported analytical redundancy relations," *IEEE Trans. Syst., Man, Cybern. A*, vol. 36, no. 6, pp. 1146–1160, 2006.  
[18] S. Ploix, A. A. Yassine, and J. M. Flaus, "An improved algorithm for the design of testable subsystems," ser. Proc. of 17th IFAC World Congress, Seoul, Korea, 2008.  
[19] B. Pulido and C. A. Gonzalez, "Possible conflicts: a compilation technique for consistency-based diagnosis," *IEEE Trans. Syst., Man, Cybern. B*, vol. 34, no. 5, pp. 2192–2206, Oct. 2004.  
[20] C. Svärd and M. Nyberg, "A mixed causality approach to residual generation utilizing equation system solvers and differential-algebraic equation theory," ser. 19th International Workshop on Principles of Diagnosis (DX-08), Blue Mountains, Australia, 2008.  
[21] L. Lovasz and M. Plummer, *Matching Theory*. AMS Chelsea Publishing, 1986.  
[22] P. Hall, "On representatives of subsets," *Journal of London Mathematical Society*, vol. 10, pp. 26–30, 1935.  
[23] A. L. Dulmage and N. S. Mendelsohn, "Covering of bi-partite graph," *Canada J. Math*, vol. 10, pp. 527–534, 1958.  
[24] K. Murota, *Matrices and Matroids for Systems Analysis*. Springer, 2000.  
[25] J. T. Pukrushpan, H. P., and A. G. Stefanopoulou, "Analysis for automotive fuel cell systems," *Transactions of the ASME*, vol. 126, pp. 14–25, 2004.  
[26] A. Rosich, R. Sarrate, and F. Nejari, "Fuel cell system benchmark," Automatic Control Department, Tech. Rep., 2008. [Online]. Available: <http://sac.upc.es>



**Albert Rosich** received the M.Sc. degrees in control and electronic engineering and the Ph.D. degree in control engineering from Universitat Politècnica de Catalunya (UPC), Terrassa, Spain, in 2005 and 2011, respectively.

He is currently with the Institut de Robòtica i Informàtica Industrial (CSIC). His current research interest include model-based fault diagnosis and structural methods for complex systems.



**Erik Frisk** was born in Stockholm, Sweden, 1971. He received the M.Sc. degree in electrical engineering and the Ph.D. degree in electrical engineering from Linköping University, Linköping, Sweden, in 1996 and 2001, respectively.

He is currently with the Department of Electrical Engineering, Linköping University. His current research interest include model-based fault detection and isolation in nonlinear large-scale systems and structural methods in fault diagnosis.



**Jan Åslund** was born in Boden, Sweden, in 1971. He received the M.Sc. degree in mechanical engineering and the Ph.D. degree in applied mathematics from Linköping University, Linköping, Sweden, in 1996 and 2002, respectively.

He is currently an Associate Professor with the Department of Electrical Engineering, Linköping University. His current research interest include model-based fault diagnosis and structural methods.



**Ramon Sarrate** received the M.Sc. degree in industrial engineering in 1994 and the Ph.D. degree in industrial engineering in 2003, both from the Universitat Politècnica de Catalunya, Terrassa, Spain.

Currently he has a position as a collaborating professor at the Department of Automatic Control at the same university. His current research interests include model based fault diagnosis and hybrid systems. He has been involved in several national and european research projects and has published several papers in scientific journals and international

conference proceedings.



**Fatiha Nejjari** received the M.Sc. degree in physics from Hassan II University and the Ph.D. degree in automatic control from Cadi Ayyad University of Marrakech in 1993 and 1997, respectively.

She is currently an Associate Professor at the Automatic Control Department of the Technical University of Catalonia (UPC). She is also a member of the Aeronautical and Space Science Research Centre (CRAE) of UPC. Her active research areas include nonlinear control, parameter and state estimation, model-based fault diagnosis, adaptive and fault tol-

erant Control. She has published several papers in journals and international events and participated in several projects and networks related with these topics.